SYSTEM, METHOD AND RECORDABLE MEDIUM FOR UPLOADING DOCUMENTS OVER A NETWORK

TECHNICAL FIELD

The present invention relates to a system, method and recordable medium for communicating over a network, and in particular, for uploading and recovering data for print jobs over a network.

BENEFIT OF EARLIER FILED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/164,723, filed November 12, 1999, U.S. Provisional Application No. 60/165,643, filed November 15, 1999, and U.S. Provisional Application No. 60/165,651, filed November 15, 1999.

BACKGROUND ART

Photocopying and reproduction services have been around since the turn of the century. These services traditionally include a variety of competition ranging from internal copy centers and network printers to outsourced and quick copy centers. These photocopying and reproduction services are used in a wide variety of commercial and non-commercial businesses. For example, publishers, firms, universities, sales professionals, training professionals, international consumers, corporations, individuals, sole proprietors, mobile offices, home offices, business travelers and content producers all utilize photocopying and reproduction services.

Internal copy centers may be found, for example, in many businesses. Internal copy centers typically handle photocopying and reproduction for personnel internal to a specific business. These copy centers are, however, often overburdened with the amount of material that they are required to produce, and are often under-staffed and limited in the number of machines that can perform the services. They are also often unreliable to users, require supervision, are often not available after certain hours and do not deliver the package after the services have been completed.

Local network printers, such as a printer attached to a personal computer, are also widely used as a photocopy and reproduction service. Local network printers often suffer from problems such as running out of toner, being in an inconvenient location, very little if any color capability, a lack of confidentiality, and no binding, stapling or shipping.

Outsourced copy centers function similar to internal copy centers except that they are run by an outside copy service. Hence, the problems associated with an internal copy center are similarly attributable to outside copy centers. In addition, the consumer must now associate with persons outside of their business. As is often the case, the exchange of information (i.e. getting the material to the outsourced service, conveying the method of binding, etc.) is miscommunicated and/or not properly conveyed. Quick copy centers (a specific kind of outsourced copy center), such as Kinko's®, Sir Speedy® and KwikCopy®, enable consumers to copy, reproduce and

bind. However, these centers have the disadvantages of inconsistent quality and service between stores, inconsistent service offerings, diluted brands, and inconvenient locations.

Fig. 1 illustrates a conventional process that a consumer must follow in order to copy or reproduce a document using, for example, the above noted copy centers. First, the consumer must save the document to disk. Then the consumer must walk to the copy center and wait in line for access to a computer and printer. An order must then be placed, and the consumer must wait for the order to be printed. Once the order has been printed, the consumer must then place the order for copy services, wait for a first copy to preview the document, confirm the order, walk back to the office, wait for copies to finish printing, walk back to the copy center and wait in line to pick up the copies. Finally, the copies must be sent to their final destination by some form of mail delivery. As explained above and by the Quick Copy Center diagram, the process is time consuming and lengthy.

As noted above, copy centers do not afford the consumer the ability to preview a document prior to completion of the service. For example, a consumer cannot view either the document as a whole or parts of the document, such as the font of the document, the binding and/or the color of the cover, until after completion of the entire photocopying and reproduction process. Hence, if the completed product is not satisfactory to the consumer, the entire process must be repeated. This not only

increases the time for copying and reproduction, but also inevitably increases the costs to both the consumer and the service provider.

SUMMARY OF THE INVENTION

In one embodiment of the invention, there is a method for uploading a document from a client to a server over a network. The method includes, for example, uploading from the client a first data packet of the document, sending to the server the first data packet, creating at the server an identification number that identifies the document, and uploading a second data packet of the document that corresponds to the identification number.

In another embodiment of the invention, there is a system for uploading a document over a network. The system includes, for example, a client that includes an upload manager that uploads and transfers the document over the network to a server, a local application that allows a user to select a print driver for printing a document, a port monitor that sends the document to the upload manager, and a print driver that receives a user selection to print a document and sends the document to the port monitor. The system also includes a server that receives the document.

In still another embodiment of the invention, there is a method for recovering data when a communication is interrupted while a document is being uploaded from a client to a server over a network. The method includes, for example, receiving a document identification code identifying the document whose transmission to the

server was interrupted, determining an interruption point indicating a portion of the document that has been transmitted successfully, and transferring to the server a remainder of the document from the interruption point.

BRIEF DESCRIPTION OF THE DRAWINGS

The figures relate to a system, method and recordable medium of the present invention. They are merely illustrative in nature and do not limit the scope of invention to their disclosed embodiments.

Fig. 1 illustrates a conventional process that a consumer must follow in order to copy or reproduce a document.

Fig. 2 illustrates an exemplary network of the invention.

Fig. 3 is an exemplary diagram of an embodiment of the system in the invention.

Fig. 4 illustrates an exemplary diagram of a user requesting a file to be printed using the system and method of the invention.

Fig. 5 illustrates an exemplary diagram of a user requesting a file to be printed using the system and method of the invention.

Fig. 6 is an exemplary diagram illustrating one embodiment of the system of the invention.

Fig. 7 is an exemplary flow diagram of a method of using the system of the invention.

Fig. 8 is an exemplary flow diagram of the operation performed by the upload manager.

Fig. 9 is an exemplary flow diagram of uploading a document.

DETAILED DESCRIPTION OF THE INVENTION

This invention supports an efficient system, method and recordable medium to copy or reproduce documents. A user may select various binding, formatting and payment options and preview the impact of such selections on a final document prior to the document being reproduced according to such selections. The system, method and recordable medium operates over a network, such as the Internet. Therefore, the user may be located anywhere in the world and request copying or reproduction of a document according to specific parameters, and may view the final document electronically before the final document is produced in a hard copy format. The user may change the document formatting and other options as desired with the preview feature. Such changes are performed in real-time from the user's computer and do not require the time and resources of a conventional copy center. Additionally, the user is not subject to waiting in line while copy center personnel handle previous requests placed by other consumers. Once an order has been placed to the reproduction system, the user can track the order to determine its status at any time during its processing.

Communication of data occurs over a network, such as a LAN, WAN, internet, or the network illustrated in the various Figures. Fig. 2 illustrates an exemplary network of the invention. Connection to the network 300 can also occur, for example, by modem or dial-up telephone connection, or as readily understood by one having skill in the art. The network 300 illustrated in the Fig. 2 includes, for example, a client 310 , a local internet service provider (ISP) 312, a universal print job acceptor ("UPJA") 320, a storage unit (e.g. a mini store) 330, and multiple web servers 315. The network 300 described in Fig. 2 is exemplary and can be modified as readily understood by one having ordinary skill in the art.

The network 300 can be divided, for explanatory purposes, into three sections: the client side of the network 300a, the back end side of the network 300b and the printing side of the network 300c. Communication between the client side 300a, the back end side 300b and the printer side 300c occurs, for example, through the network 300. As illustrated in Fig. 2, added levels of security, such as use of firewalls, ensure that information sent over the network 300 is not disturbed (e.g. the information is not modified, changed or breached). Generally speaking, a user on the client side 300a of the network can request printing from, for example, a personal computer, and generate a document for shipping and/or delivery from printer side 300c. The back end side 300b of the network 300b is transparent to the user.

Fig. 3 is an exemplary diagram of an embodiment of the system in the present invention. Fig. 3 primarily illustrates the client side 300a of the network 300, with

the addition of the UPJA and database, such as a SQL server, which are on the back end side 300b of the network 300. The system includes, for example, a Data Store, a Print Driver UI, a local application, a graphics library, a Print Driver PDL Gen, a Print Spooler, a Port Monitor, an Upload Manager, a Universal Print Job Acceptor (UPJA), a SQL server, a Version Manager, and a System Tray Application which allows pop-up menus and user selection of icons.

First, relevant file information (parameters) will be sent from the spooler 344 via the port monitor 346 and upload manager 310a to a web server 320 (e.g. UPJA). The relevant file information includes, for example, a handle identifying the location of the printer, printer name, job id, printing level, and document information such as color, stapling, etc. After the relevant file information is sent to the port monitor 346, a check is performed to determine whether a valid component (e.g. a print driver 310b) is being used, for example, to print the document. In the preferred embodiment, the print driver resides as a file on the client, and comprises Windows™ based code. The print driver 310b may be installed, for example, by downloading it from a server or installed from disk. In order to validate the print driver 310b, the identity of the print driver 310b is sent to the version manger 348, and a check (e.g. a CRC) is performed to compare the components of the print driver stored in the file with information stored in a registry. The registry includes, for example, predetermined information which can authenticate that a valid print driver is being used. If the comparison results in an invalid print driver, the data file (document) will

not be sent (uploaded) to the web server. If, on the other hand, the comparison results in a determination that the component (print driver) is valid, then the data file is sent from the print spooler 344 to the port monitor 346. The data file is sent, for example, as packets of information from the print spooler 344 to the port monitor 346. In this embodiment, it is preferable that the packets of data are sent in 4 Kbyte packets. Once the data file (either as a whole or as individual packets) has been sent to the port monitor 346, the data file or packet is sent, via the upload manager 310c, to the UPJA 320 and stored. The upload manager 310c then launches a web browser 310d for viewing the document.

In one embodiment of the invention, an object (e.g. a filter) may be placed in between two other objects (e.g. the Print Spooler and the Port Monitor). The filter can be used, for example, to detach the port monitor 346 from the Print Spooler 344, allowing the system to easily upgrade object code for future versions of software. The filter would preferably be a proxy of either one of the two objects. Other embodiments may include a data store filter in between the DevMode Data Store and the Print Driver UI, and a Print Driver Filter in between the GDI and the Print Driver PDL Gen.

With reference to Figs. 2 and 3, the client 310 (e.g. a terminal, personal computer, PDA, mobile phone, etc.) stores, for example, local applications 310a such as Word ™ or PowerPoint™, print drivers 310b, a port monitor 346, an upload manager 310c and a browser 310d. The local applications 310a can be used to create

or download a document (the term "document" is being used to broadly refer to any data or information that can be transmitted over the network 300) that the user can ultimately forward to the print side 300c for shipping and/or delivery to a specified location. Print driver 310b builds and creates objects necessary to communicate with the selected printing device (e.g. a printer directly attached to the client 310, a printer on the network 300 or a printer located at the printer side 300c). The upload manager 310c is responsible for compressing and transferring files (e.g. documents or selections of documents) over the network. The browser 310d, such as Internet Explorer ™ or Netscape Navigator ™, is used to download client software (e.g. print drivers) 310b, and to view and order documents. Downloading of this data preferably occurs prior to requesting a print job. Of course, other systems and methods may be used to browse and download print drivers and view or order documents as one having ordinary skill in the art would recognize. For example, print drivers 310b may be loaded onto the client 310 by reading software stored on a recordable medium. The upload manager 310c communicates with the UPJA 320, for example, via Extensible Markup Language (XML) messages over http/https. XML allows browser clients to download an HTML page and then manipulate the page off line, without referring back to the server. The main task of the upload manager 310c and UPJA 320 is to transfer and compress files (preferably secure pdl files) via, for example, a Secure Socket Layer (SSL).

Operation of the upload manager 310c is now described. The upload manager 310c, which resides on the client 310 and is software in the preferred embodiment, handles upload and recovery of data for print jobs on the network 300. Once a complete document has been uploaded and transmitted to a server for reproduction processing, the upload manager 310c launches a web browser that allows a user to view and edit the document. For software including version information, the upload manager 310c can determine which version of the software is supported by the system and can monitor the data to determine whether a valid version of the software is being used, as described below.

A caching server 340 manages the download of driver software and other common HTML and image data elements between the client 310 and the UPJA 320. The UPJA 320 (which is a server in the preferred embodiment and therefore includes the conventional components of a computer, including memory, storage and a process, in the preferred embodiment), in one embodiment, receives the document as it is transmitted from the client 310, via the ISP 312, across the network 300. The document is received by the UPJA 320, via the upload manager 310c, after the print driver 310b being used has been authenticated by the version manager 348, and the entire document has been sent to the port monitor 346. Replicating servers 350 (i.e. lightweight directory assistance protocol, or LDAP, servers) authenticate clients 310 requesting services from the UPJA 320 and web servers 315. Authentication occurs as is readily understood by one having ordinary skill in the art. For example, the web

servers 315 provide application services for web browsing and viewing of order information of newly uploaded and previously ordered documents. Software, such as AlchemyPS, is used to render images of the PostScript document for viewing. The mini store 330 stores documents (i.e. files) uploaded to the UPJA 320, as well as web, application, SQL and LDAP data.

After a document has been sent to the UPJA 320, it can be downloaded to the printer side 300c. The printer side 300c allows the user on the client side 300a to print, bind and deliver documents that have been uploaded and stored on the network 300. The physical location of the print side 300c can be anywhere relative to the client side 300a and back end 300b. In the preferred embodiment, print side 300c is located in a printing facility next door to a delivering company such as Federal Express™. Documents are downloaded to a storage unit (e.g. main storage) located on print side 300c, as illustrated in Fig. 2, and then replicated on printers for ultimate shipping and delivery of the completed product to an address or location specified by the user. The completed product is an actual representation of the virtual product created by the user on the client side 300a. For example, a user may select a file that has been created using a standard editor such as Word ™. The file may then be edited or modified by selecting font size and color, binding and paper using the interface opened by browser 310d. More specifically, Figs. 4 and 5 illustrate exemplary diagrams of a user requesting a file to be printed using the system and

method of the present invention. Operation of the system and method are discussed below.

Referring to Fig. 4, a user requests data to be printed at 400. After data (i.e. a document) has been selected at 400, the user may configure the settings of the document using a configuration unit (e.g. the item configuration wizard), as described below. Before uploading the selected document(s) to the UPJA 320, the print driver 310b selected by the user is verified by the version manager 348 at 410. The version manager is preferably a separate component that is accessed to verify that all of the client components being used by the client are valid before any uploading occurs. If the version manager 348 determines that the print driver is not valid (i.e. not acceptable), then a message is sent to the client 310 at 450. If, on the other hand, the version manager 348 determines that the print driver information is valid, then data is uploaded from the client 310, for example, to the port monitor 346 via the print spooler 344 (see, for example, Fig. 37. That is, when data being uploaded includes version information, the upload manager 310c can query about, for example, what version of software the system supports. For example, when a "print" command is issued, the upload manager 310c can monitor, in real time, the data to determine if a valid version of the print driver 310b (e.g. driver software) is being used. The upload manager 310c also verifies that the client 310 software (e.g. local applications), and/or components making up the software, represent a coherent set of components

and work properly together. This can be accomplished, for example, by a version manager 348 authenticating the components of the drivers and client software.

Uploading a document begins at 420. The upload manager 310c compresses the document(s) that has been selected for printing, and sends it to the UPJA 320 for storage (preferably temporary) at 430. The preferable method of transferring documents using the upload manager is the discussion of a separate application. Of course, one having ordinary skill in the art will recognize that documents can be transferred in a variety of ways, and the present invention is not limited to the preferred embodiment. Documents sent to UPJA 320 may also be stored in mini store 330 for later retrieval at 430. Finally, the document is sent to printer side 300c where it is stored in a main storage unit, and printed according to the user defined specifications at 440.

As illustrated in Fig. 5, subsequent to uploading a document in 420, in one embodiment users can specify with the aid of a configuration wizard, the item, for example, document type, paper stock, printing options, double or single sided copies, color versus black and white, cover and orientation (422). The user is then presented with the option of either viewing their shopping cart and choosing an item (e.g. another document) to be purchased for printing, or can select the number of copies to be printed (422). After the selection of the number of copies has been made, the user can select the appropriate address or location for delivery of the printed document,

convey billing information, preview and submit the order, and receive confirmation and updates regarding the order (424 and 442).

Fig. 6 is an exemplary diagram illustrating one embodiment of the system of the invention. The network 300 includes, for example, client side 300a, back end side 300b and print side 300c. The client side 300a includes a public web server, third party content, a corporate intranet and a print driver. Each of these components can freely communicate with each other and with the UPJA 320 on back end side 300b. The UPJA 320, as described above, can receive jobs, requests for information, etc., and serves as the primary (although not the exclusive) link between the client side 300a and the print side 300b. The back end side 300c includes a main storage to store documents downloaded from the UPJA 320, a job queue, workstations and printers to complete the ordered jobs. Servicing on the completed jobs can also be handled at the print side 300c, or at a separate location. Servicing includes, for example, binding, cutting, collating and wrapping the documents to be shipped, as well as boxing, labeling and shipping or delivery of the document(s) to a specified address or location.

Fig. 7 is an exemplary flow diagram of a method of using the system in the invention. The flow illustrated is merely an example of one embodiment of a process that a user accessing the system may perform. The user creates a document in a local application 310a on, for example, their personal computer (600), and prints the documents from the application to the selected print driver 310b (605). The document

is printed as a postscript document (610). The upload manager 310c then uploads the postscript print file to the UPJA 320 over http/https (615), and the upload manager launches the web browser 310d for document viewing (620). The user can then configure the finishing and binding options for the document using the interface on the personal computer (625). Once document configuration information is validated, the user inputs shipping and payment data on the interface(630). The shipping and payment data are verified, and the print file is put in long term storage (635). The finishing and biding options are then combined with the postscript file to create a print ready file (640), and the print ready file is sent to the print queue (645) and transferred to the production facility (i.e. printing facility). A printer operator can then select a job and queues it to an available printer (655), and the job is ripped and sent to the printer (660). The printer punches and/or binds the job on-line (665), and the package is sent for delivery (675). The user is able to track the package, receive the package and open the package (680-690).

Fig. 8 is an exemplary flow diagram of the operation performed by the upload manager. A client's "upload request" is sent via, for example, an XML protocol such as a Java active server page (ASP) to a server, such as the UPJA 320 depicted in Fig. 3. The server creates an "upload.begin" request ASP and an associated object identifying the request. The object is sent to a database and stored. In response, the object initiates an "upload.begin" response, which is transmitted to the client 310 via, for example, XML. This process repeats for each client upload request. Further

details of the processing performed relative to uploading an entire document are described below relative to Fig. 5.

Fig. 9 is an exemplary flow diagram of uploading a document. Referring to Fig. 9, the transfer of information across the network 300 begins with a request by the client 310, for example, a command to print a file at 506. The print driver 310b residing, for example, on the client 310 makes a request to upload a data packet at 510. The data packet is sent as an ASP request to web server 315, such as a IIS Windows-based web server at 516. The request may be, for example, an "upload-begin" request, an "upload-request" or an "upload-complete-response."

The ASP provides the server-side equivalent to using a scripting language and objects on the client 310. When the print driver 310c makes the request to upload a data packet, the client 310 keeps track of which data packet(s) has been sent. That is, the data packet is "marked" for subsequent reference. If a problem occurs in the transmission, for example a modem failure, machine failure, the internet goes down, a database goes down, the file is corrupted, etc., or if the file is successfully transmitted, the system can properly respond to the client 310. For example, if the file is successfully transmitted, the client 310 is notified of the successful transfer. If, on the other hand, a transmission error occurs, the network 300 is able to identify which data packet was last sent using the marked reference. In this regard, the system can continue, i.e. resume, uploading from the point of error, without having to resend or retransmit the entire data file.

The ASP request then executes at the server 315 and launches associated objects stored in the server at 520. After the ASP request executes, an initial handshake occurs between the client 310 and server 315. During the initial handshake, an object is launched to create a new globally unique identification (GUID) of the file at 526. The newly created GUID is then sent to the client 310 as an "upload-begin-response" and is used for subsequent "upload-request" uploading of the file. Uploading of the file then begins and information about the file, such as file size, CRC, document names, color, orientation and number of pages, is transferred to the UPJA 320 at 532. Once the entire file and corresponding file information is transferred to and stored in the UPJA 320 at 538, it is transferred and stored in a computer readable medium, such as, a mini store 330 at 542.

If at any time during the transfer of data an error occurs, an "upload-response" is sent to the client. The response indicates to the client 310 that an error during transmission has occurred at 550. The client 310 responds to the "upload-response" by re-transmitting the packets from a point at which the last packet was successfully sent at 556. This re-transmission process can be repeated for a predetermined number of times. If an error is still present after the system has re-tried the predetermined number of times at 562, then an "upload-resume" request is initiated at 568. The "upload-resume" request transmits the GUID to the upload manager 310c so that the upload manager 310c can determine which file the upload resume request relates to. The upload manager 310c then determines the last referenced, or "marked," data

packet and resumes the transfer of data beginning with that packet. The data packets are transmitted until the entire file has been successfully transferred to the server.

Once the complete file, along with corresponding file information, has been transferred to the server and stored, a browser 310d located at the client 310 is launched at 574. The browser 310d allows the user to view, configure, format, print, etc. the uploaded document as described above.

Although the invention has been described relative to a particular embodiment, one of skill in the art will appreciate that this description is merely exemplary and the system and method of this invention may include additional or different components. This description is therefore limited only by the appended claims and the full scope of their equivalents.